

АНАЛІЗ ПРОДУКТИВНОСТІ МІКРОСЕРВІСНИХ СИСТЕМ НА .NET

Фомічов М.О., Федорченко В.М.

Харківський національний економічний університет імені Семена Кузнеця,
Харків, Україна

Оцінка ефективності мікросервісних архітектур є ключовим аспектом їх розробки та експлуатації. Зростання корпоративних інформаційних систем вимагає рішень, що забезпечують високу продуктивність, масштабованість та надійність. Наприклад, дослідження показують, що мікросервісна архітектура суттєво впливає на підтримуваність і масштабованість програмних систем [1]. Інші праці демонструють, що вибір комунікаційної технології між сервісами (REST, gRPC) має значний вплив на продуктивність і використання ресурсів [2]. Мікросервісна архітектура, побудована за допомогою фреймворку ASP.NET Core та платформи .NET 9, має широке поширення завдяки можливостям швидкого розгортання, ефективного масштабування та модульного управління. Зокрема, результати досліджень підтверджують, що такі підходи дозволяють створювати більш гнучкі, підтримувані системи [3]. Однак для досягнення справді високої продуктивності важливо враховувати низку ключових параметрів: комунікаційний протокол між службами, розмір і структура payload-запитів, налаштування середовища виконання, ізоляцію ресурсів та конфігурацію контейнерів чи віртуальних машин.

В доповіді окрема увага приділяється аналізу способів комунікації між сервісами — через HTTP-запити (REST/gRPC) та через брокери повідомлень (наприклад, RabbitMQ чи Apache Kafka). Різні дослідження показують, що технологія передачі та протокол можуть суттєво впливати на затримку, пропускну здатність та використання ресурсів [2]. Тестування продуктивності мікросервісних систем найчастіше здійснюється із застосуванням інструментів навантажувального тестування, таких як Apache Benchmark, k6 та Apache JMeter [4]. Ці інструменти дозволяють зібрати метрики: latency (затримка), throughput (пропускну здатність), CPU/memoгу використання, стійкість під навантаженням. Відповідно до сучасних досліджень, важливо поєднувати короткочасні пікові тестування з тривалими стрес-тестами, щоб отримати повну картину продуктивності системи [1]. У подальших дослідженнях рекомендується провести порівняльний аналіз існуючих підходів до тестування з метою визначення найефективнішого методу оцінки продуктивності мікросервісних систем. Для практичної перевірки обраних методів використовується відкритий (опенсорсний) мікросервісний застосунок, який буде реалізований на базі ASP.NET Core / .NET 9. Додаток має бути протестовано за допомогою обраних інструментів – Apache Benchmark, k6 та JMeter. Отримані результати дозволять визначити оптимальний підхід до оцінки продуктивності мікросервісної архітектури у реальних умовах. Розвиток мікросервісної архітектури є важливим напрямком оптимізації корпоративних інформаційних систем. Збільшення обсягів оброблюваних даних та необхідність високої швидкості обробки вимагають не лише нових технологій, а й розробки чітких критеріїв оцінки продуктивності. Оскільки мікросервіси на

платформах ASP.NET Core / .NET 9 дозволяють створювати високопродуктивні системи з ефективним управлінням ресурсами, простотою інтеграції та зручним розгортанням — надзвичайно важливо проаналізувати різні конфігурації системи та знайти найкращі практики.

Список літератури

1. Jarmoszewicz, J., Iwanowski, P., & Plechawska-Wójcik, M. (2024). Analysis of the performance and scalability of microservices depending on the communication technology. *Journal of Computer Sciences Institute*, 33, 323-330. DOI: <https://doi.org/10.35784/jcsi.6499>
2. Niswar, M., Safruddin, R. A., Bustamin, A., & Aswad, I. (2024). Performance Evaluation of Microservices Communication with REST, GraphQL, and gRPC. *Int. Journal of Electronics and Telecom.*, 70(2). DOI: <https://doi.org/10.24425/ijet.2024.149562>
3. Krekora, K. (2024). Performance analysis of .Net and Spring Boot microservices on Microsoft Azure. *Journal of Computer Sciences Institute*, 33, 258-263. DOI: <https://doi.org/10.35784/jcsi.6268>
4. Urias, I., & Rossi, R. (2023). Evaluation of Frameworks for MLOps and Microservices. *EAI Endorsed Transactions on Smart Cities*, 7(3). DOI: <https://doi.org/10.4108/eetsc.3661>

ПАРАЛЕЛЬНІ АЛГОРИТМИ ОБРОБКИ ВЕЛИКИХ ДАНИХ У РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СЕРЕДОВИЩАХ

Ні Я.С., Ні О.В., Шостак М.В., Шостак В.В.

Харківський національний університет радіоелектроніки, Харків, Україна

У сучасному світі обсяг даних, що генерується в різних сферах діяльності, від фінансових систем до телекомунікаційних мереж і наукових досліджень, зростає експоненційно. Це створює необхідність у застосуванні ефективних методів обробки великих даних, здатних забезпечити швидке і точне отримання результатів. Одним із найбільш перспективних підходів є використання паралельних алгоритмів у розподілених обчислювальних середовищах, таких як кластери, хмарні платформи та суперкомп'ютери.

Паралельна обробка великих даних дозволяє розподіляти обчислювальні задачі між кількома вузлами системи, що значно підвищує продуктивність і скорочує час обробки. Сучасні фреймворки, такі як Apache Hadoop, Apache Spark та Dask, надають можливість реалізувати алгоритми паралельного виконання, зокрема для задач сортування, агрегації, аналізу потоків даних та машинного навчання. Розподілена обробка також дозволяє підвищити відмовостійкість системи: у випадку виходу з ладу одного вузла обчислення можуть бути автоматично перенаправлені на інші вузли без втрати даних. Особливо важливим є застосування паралельних алгоритмів у сфері аналітики великих даних (Big Data Analytics). Завдяки можливості одночасного опрацювання великих масивів даних можна реалізувати складні алгоритми прогнозування, виявлення закономірностей, кластеризації та пошуку аномалій. Це дозволяє оперативнo отримувати корисну інформацію з даних, яка має критичне значення для прийняття рішень у бізнесі, науці та державному управлінні.